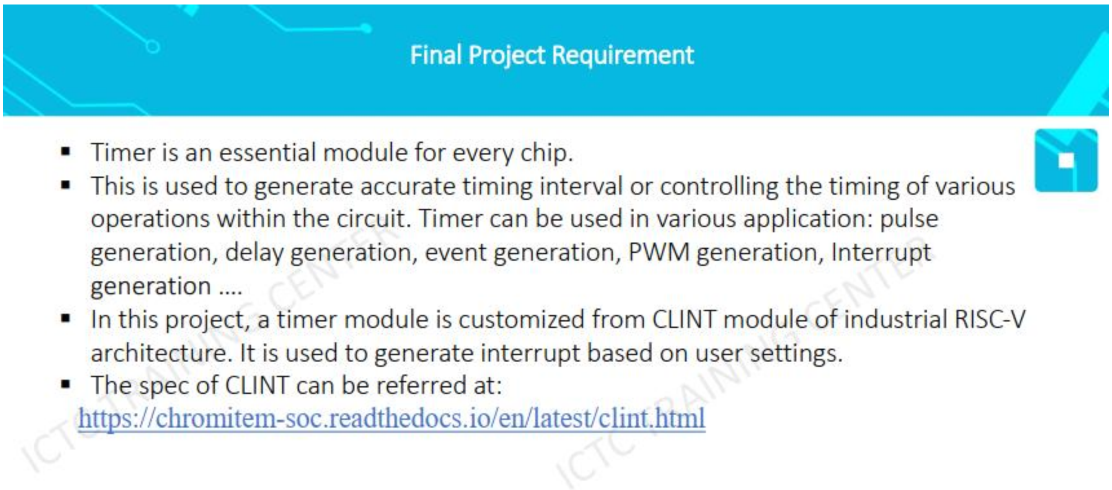
1. Overview



Final Project Requirement

- Timer is an essential module for every chip.
- This is used to generate accurate timing interval or controlling the timing of various operations within the circuit. Timer can be used in various application: pulse generation, delay generation, event generation, PWM generation, Interrupt generation ....
- In this project, a timer module is customized from CLINT module of industrial RISC-V architecture. It is used to generate interrupt based on user settings.
- The spec of CLINT can be referred at: https://chromitem-soc.readthedocs.io/en/latest/clint.html

2. Functional Requirements

2.1 General Features

- 64-bit count-up timer.
- APB Slave interface for register configuration. 32bit trans
- Address space: 4KB (0x4000_1000 – 0x4000_1FFF).
- System clock frequency: 200 MHz.
- Active low asynchronous reset (`sys_rst_n`).

2.2 Counting Modes

- **Default mode:** The counter increments based on the system clock.

- **Control mode:** Counter speed can be adjusted using a divisor value (`div_val`) when the control mode is enabled (`div_en`).

(TCR.div_en ==1) - TCR [1]

(TCR.div_val == n) - TCR [11:8]

- **Halted mode**: The timer can be halted in debug mode. The halt is triggered when `debug_mode` is high and a halt request (`THCSR.halt_req`) is asserted. The timer resumes when the halt request is cleared.

(Debug_mode ==1)

(THCSR.halt_req ==1) - THCSR [0]

- **Error handling mode**: In control mode, changes to the divisor settings (`div_val`) while the timer is running results in an error response.

(TCR.div_val changed while running)

(TCR.div_en changed while running)

(TCR.div_val loaded with invalid input)

- **Byte access**: The bus supports byte access to the individual registers. ????

- **Wait states**: A 1-cycle wait state can be inserted during APB transactions to improve timing.

2.3 Timer Interrupt

- Interrupt generation: An interrupt (`tim_int`) is triggered when the counter matches the compare value (`TCMP0` and `TCMP1`).
- Interrupt can be enabled or disabled via the `TIER` register.

(TIER [0]==1)

- Interrupt status can be cleared by writing to the `TISR.int_st` bit.

(TISR [0]==1)

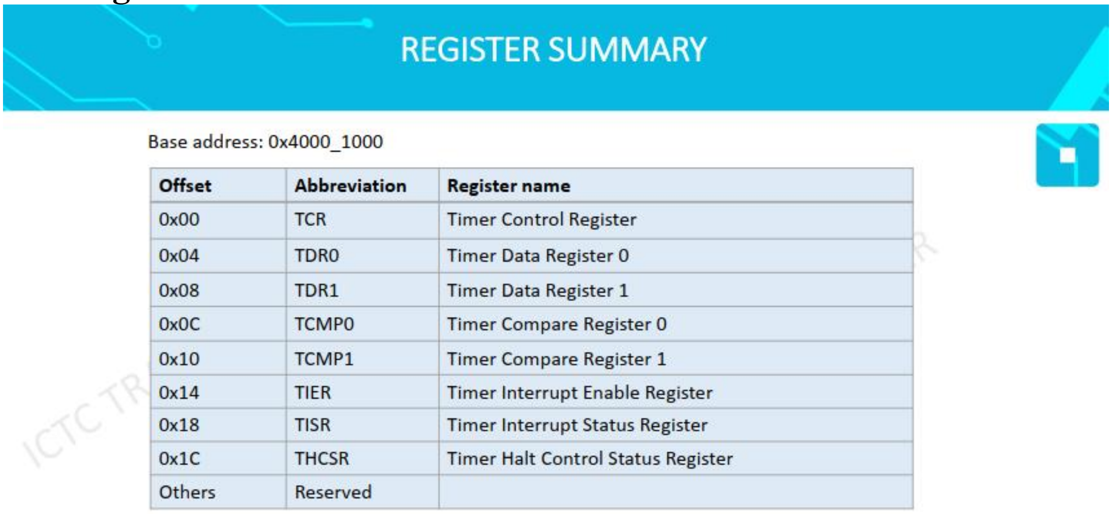
3. Design Details

3.1 APB Slave Interface

The Timer IP communicates with an external master device via the APB interface. The APB interface includes the following signals:

- tim_psel: Select signal for the timer.
- tim_pwrite: Write enable signal.
- tim_penable: APB enable signal.
- tim_paddr: 32-bit address bus.
- tim_pwdata: 32-bit write data bus.
- tim_prdata: 32-bit read data bus.
- tim_pready: Ready signal indicating the transfer is complete.
- tim_pslverr: Error response signal.

3.2 Register Set

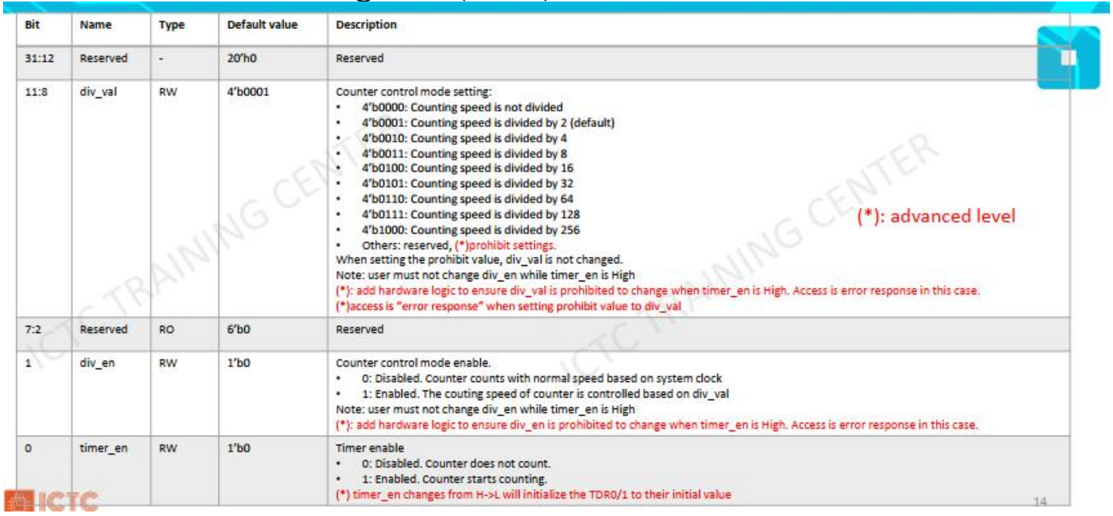


REGISTER SUMMARY

Base address: 0x4000_1000

Offset	Abbreviation	Register name
0x00	TCR	Timer Control Register
0x04	TDR0	Timer Data Register 0
0x08	TDR1	Timer Data Register 1
0x0C	TCMP0	Timer Compare Register 0
0x10	TCMP1	Timer Compare Register 1
0x14	TIER	Timer Interrupt Enable Register
0x18	TISR	Timer Interrupt Status Register
0x1C	THCSR	Timer Halt Control Status Register
Others	Reserved	

3.3 Timer Control Register (TCR)



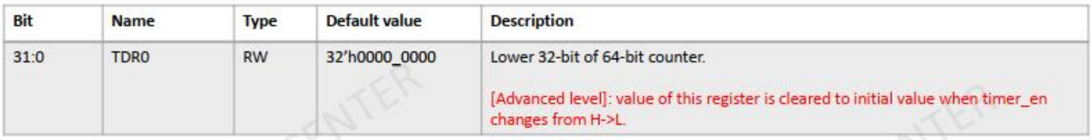
Bit	Name	Type	Default value	Description
31:12	Reserved	-	20'h0	Reserved
11:8	div_val	RW	4'b0001	Counter control mode setting: • 4'b0000: Counting speed is not divided • 4'b0001: Counting speed is divided by 2 (default) • 4'b0010: Counting speed is divided by 4 • 4'b0011: Counting speed is divided by 8 • 4'b0100: Counting speed is divided by 16 • 4'b0101: Counting speed is divided by 32 • 4'b0110: Counting speed is divided by 64 • 4'b0111: Counting speed is divided by 128 • 4'b1000: Counting speed is divided by 256 • Others: reserved, (*)prohibit settings. When setting the prohibit value, div_val is not changed. Note: user must not change div_en while timer_en is High (*): add hardware logic to ensure div_val is prohibited to change when timer_en is High. Access is error response in this case. (*)access is "error response" when setting prohibit value to div_val
7:2	Reserved	RO	6'b0	Reserved
1	div_en	RW	1'b0	Counter control mode enable. • 0: Disabled. Counter counts with normal speed based on system clock • 1: Enabled. The couting speed of counter is controlled based on div_val Note: user must not change div_en while timer_en is High (*): add hardware logic to ensure div_en is prohibited to change when timer_en is High. Access is error response in this case.
0	timer_en	RW	1'b0	Timer enable • 0: Disabled. Counter does not count. • 1: Enabled. Counter starts counting. (*) timer_en changes from H->L will initialize the TDR0/1 to their initial value

(*): advanced level

3.4 Timer Data Registers (TDR0/TDR1)

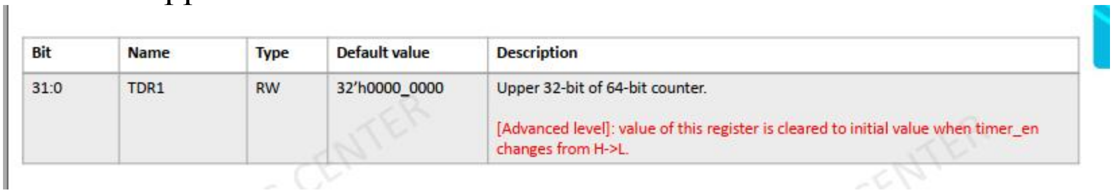
These registers store the 64-bit counter value.

- TDR0: Lower 32 bits of the 64-bit counter.



Bit	Name	Type	Default value	Description
31:0	TDR0	RW	32'h0000_0000	Lower 32-bit of 64-bit counter. [Advanced level]: value of this register is cleared to initial value when timer_en changes from H->L.

- TDR1: Upper 32 bits of the 64-bit counter.

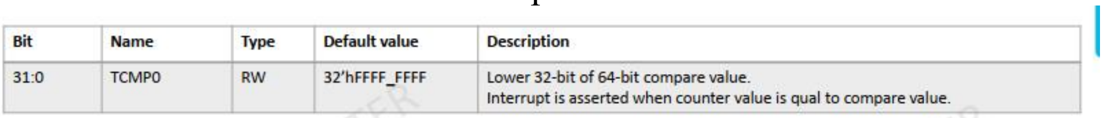


Bit	Name	Type	Default value	Description
31:0	TDR1	RW	32'h0000_0000	Upper 32-bit of 64-bit counter. [Advanced level]: value of this register is cleared to initial value when timer_en changes from H->L.

3.5 Timer Compare Registers (TCMP0/TCMP1)

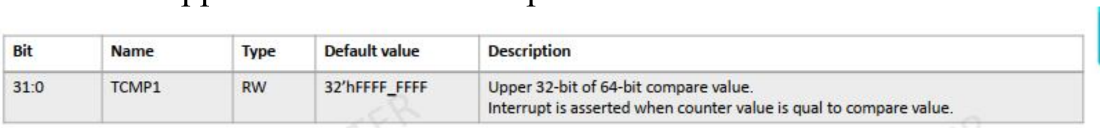
These registers hold the 64-bit compare value. When the counter reaches this value, an interrupt is generated.

- TCMP0: Lower 32 bits of the compare value.



Bit	Name	Type	Default value	Description
31:0	TCMP0	RW	32'hFFFF_FFFF	Lower 32-bit of 64-bit compare value. Interrupt is asserted when counter value is qual to compare value.

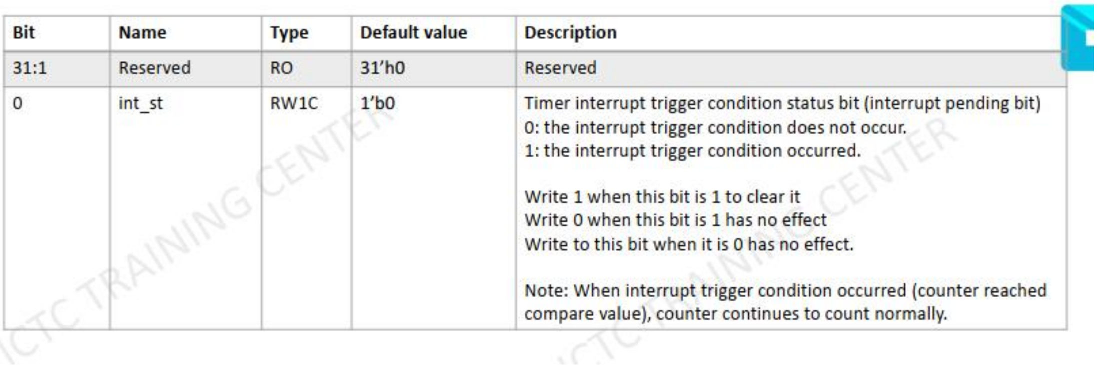
- TCMP1: Upper 32 bits of the compare value.



Bit	Name	Type	Default value	Description
31:0	TCMP1	RW	32'hFFFF_FFFF	Upper 32-bit of 64-bit compare value. Interrupt is asserted when counter value is qual to compare value.

3.6 Timer Interrupt Enable Register (TIER)

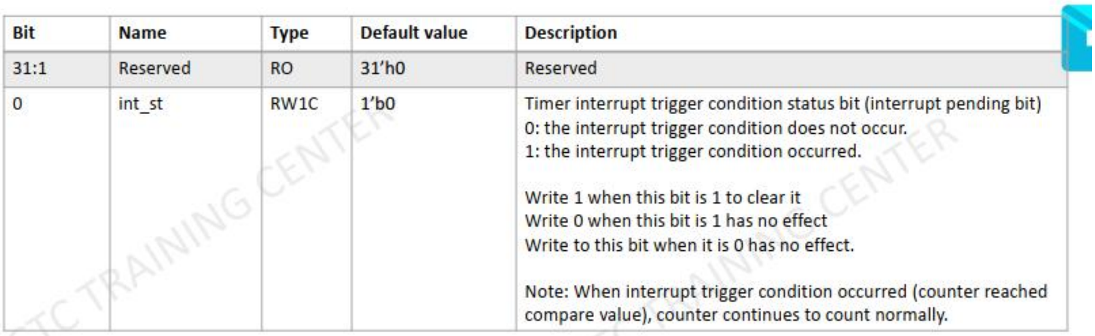
This register controls whether the timer interrupt is enabled.



Bit	Name	Type	Default value	Description
31:1	Reserved	RO	31'h0	Reserved
0	int_st	RW1C	1'b0	Timer interrupt trigger condition status bit (interrupt pending bit) 0: the interrupt trigger condition does not occur. 1: the interrupt trigger condition occurred. Write 1 when this bit is 1 to clear it Write 0 when this bit is 1 has no effect Write to this bit when it is 0 has no effect. Note: When interrupt trigger condition occurred (counter reached compare value), counter continues to count normally.

3.7 Timer Interrupt Status Register (TISR)

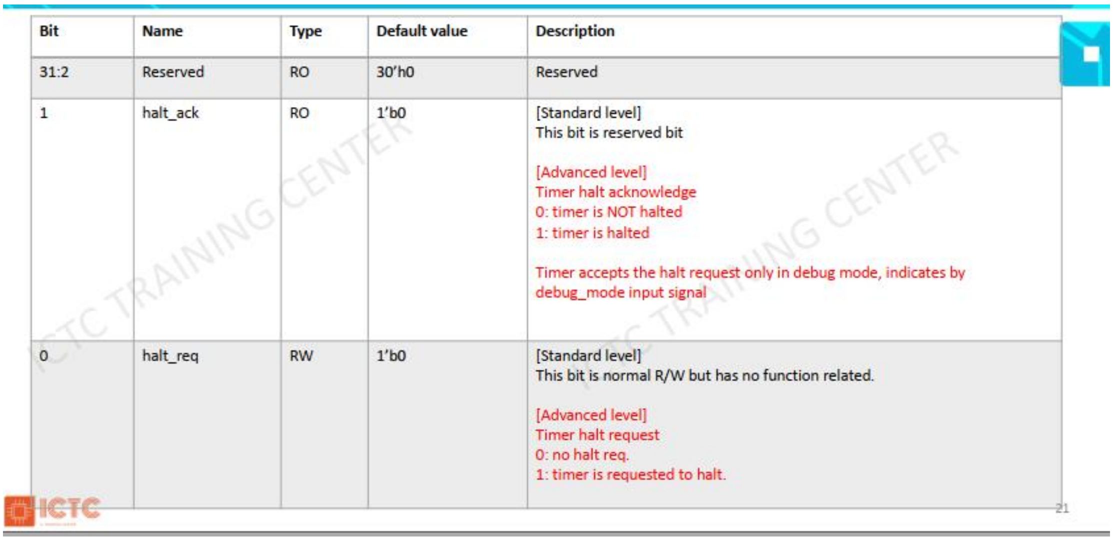
This register indicates whether an interrupt has occurred. It must be cleared by software after the interrupt is handled.



Bit	Name	Type	Default value	Description
31:1	Reserved	RO	31'h0	Reserved
0	int_st	RW1C	1'b0	Timer interrupt trigger condition status bit (interrupt pending bit) 0: the interrupt trigger condition does not occur. 1: the interrupt trigger condition occurred. Write 1 when this bit is 1 to clear it Write 0 when this bit is 1 has no effect Write to this bit when it is 0 has no effect. Note: When interrupt trigger condition occurred (counter reached compare value), counter continues to count normally.

3.8 Timer Halt Control Status Register (THCSR)

This register controls the halt mode operation of the timer.



Bit	Name	Type	Default value	Description
31:2	Reserved	RO	30'h0	Reserved
1	halt_ack	RO	1'b0	[Standard level] This bit is reserved bit [Advanced level] Timer halt acknowledge 0: timer is NOT halted 1: timer is halted Timer accepts the halt request only in debug mode, indicates by debug_mode input signal
0	halt_req	RW	1'b0	[Standard level] This bit is normal R/W but has no function related. [Advanced level] Timer halt request 0: no halt req. 1: timer is requested to halt.

4. Signal List

Top module : timer_top

Signal name	**Width**	**Direction**	**Description**	**Type**
Sys_clk	1	input	System clk	wire
Sys_rst_n	1	input	Reset sig	wire
Tim_psel	1	input	APB select signal for time	wire
tim_pwrite	1	input	APB write enable sig	wire
Tim_penable	1	input	APB enable sig	wire
Tim_addr	32	input	APB address bus for reg select	wire
Tim_pwdata	32	input	APB write data bus	wire
Tim_prdata	32	output	APB read data bus	wire
Tim_pready	1	output	APB ready sig	wire
Tim_pslverr	1	output	APB error sig	wire
Tim_int	1	output	Time interrupt sig	wire
Dbg_mode	1	output	Debug dev mode	wire

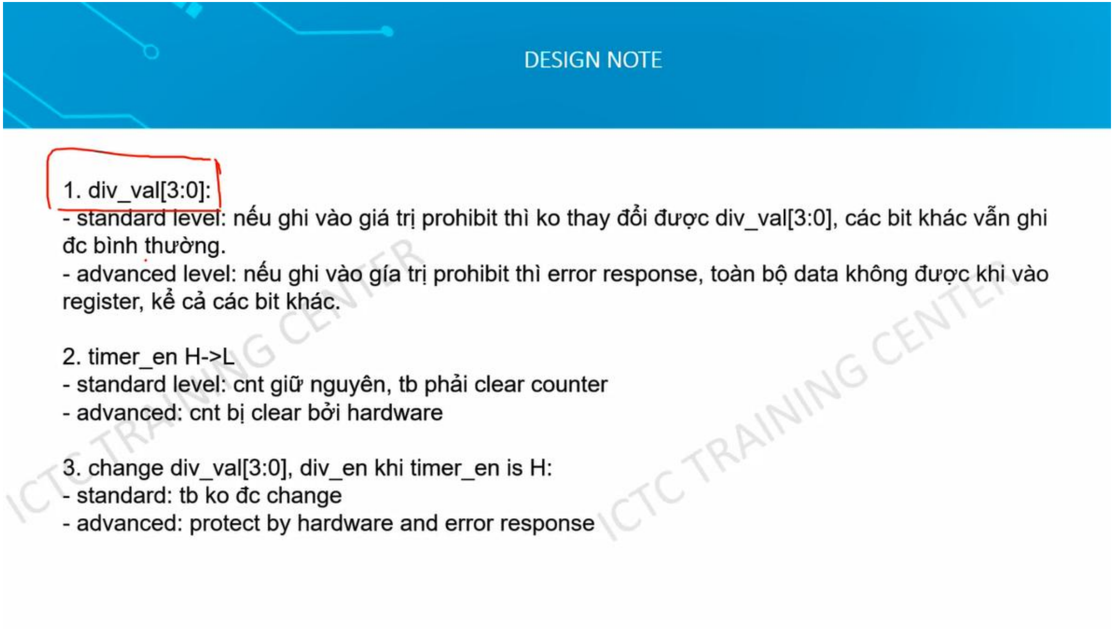
Module timer_ctrl

Signal name	**Width**	**Direction**	**Description**	**Type**
Sys_clk	1	input	System clk	wire
Sys_rst_n	1	input	Reset sig	wire
Timer_en	1	input	Timer enable sig	wire
Div_en	1	input	Division enable sig	wire
Div_val	4	input	Division val sig	wire
Tcmp	64	input	Timer data reg	wire
Halt_req	1	input	Halt sig	wire
Int_en	1	input	Interrupt enable sig	wire
Dbg_mode	1	input	Dev mode enable sig	wire
tdr	64	output	Timer data reg	reg
Halt_ack	1	output	Halt acknowledge sig	reg
Int_st	1	output	Interrupt status sig	reg
Tim_int	1	output	Timer interrupt sig	reg

Module APB

Signal name	**Width**	**Direction**	**Description**	**Type**
Sys_clk	1	input	System clk	wire
Sys_rst_n	1	input	Reset sig	wire
Tim_psel	1	input	Timer enable sig	wire
Tim_pwrite	1	input		wire
Tim_penable	1	input		wire
Tim_paddr	32	input		wire
Tim_pwdata	32	input		wire
Tim_prdata	32	input		wire
Halt_ack	1	input		wire
Int_st	1	input		wire
Tim_prdata	32	output		reg
Tim_pready	1	output		reg
Tim_pslverr	1	output		reg
Timer_en	1	output		reg
Div_en	1	output		reg
Div_val	4	output		reg
Tdr	64	output		reg
Tcmp	64	output		reg
Halt_req	1	output		reg
Int_en	1	output		reg

Design note

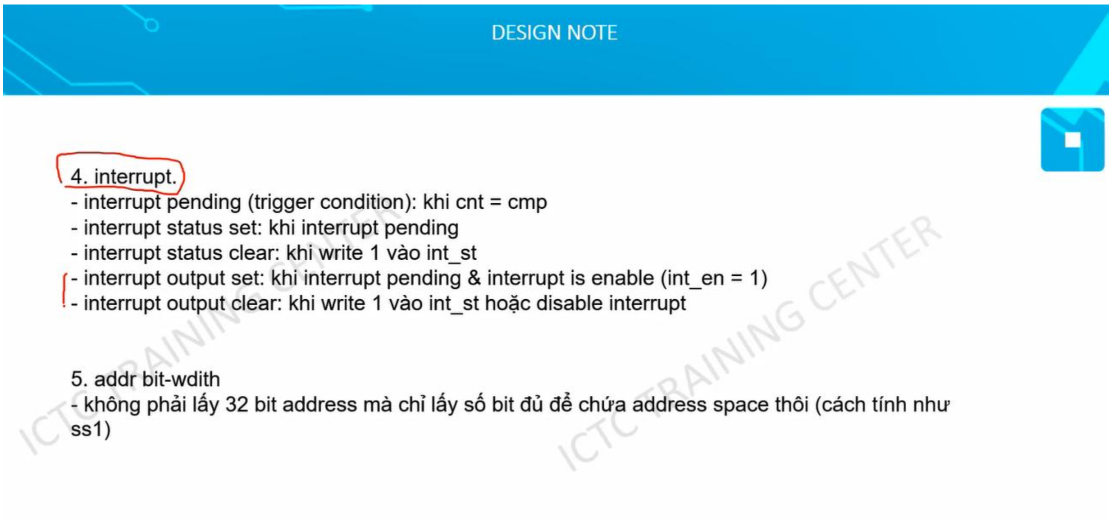


DESIGN NOTE

1. div_val[3:0]:
- standard level: nếu ghi vào giá trị prohibit thì ko thay đổi được div_val[3:0], các bit khác vẫn ghi đc bình thường.
- advanced level: nếu ghi vào gía trị prohibit thì error response, toàn bộ data không được khi vào register, kể cả các bit khác.

2. timer_en H->L
- standard level: cnt giữ nguyên, tb phải clear counter
- advanced: cnt bị clear bởi hardware

3. change div_val[3:0], div_en khi timer_en is H:
- standard: tb ko đc change
- advanced: protect by hardware and error response



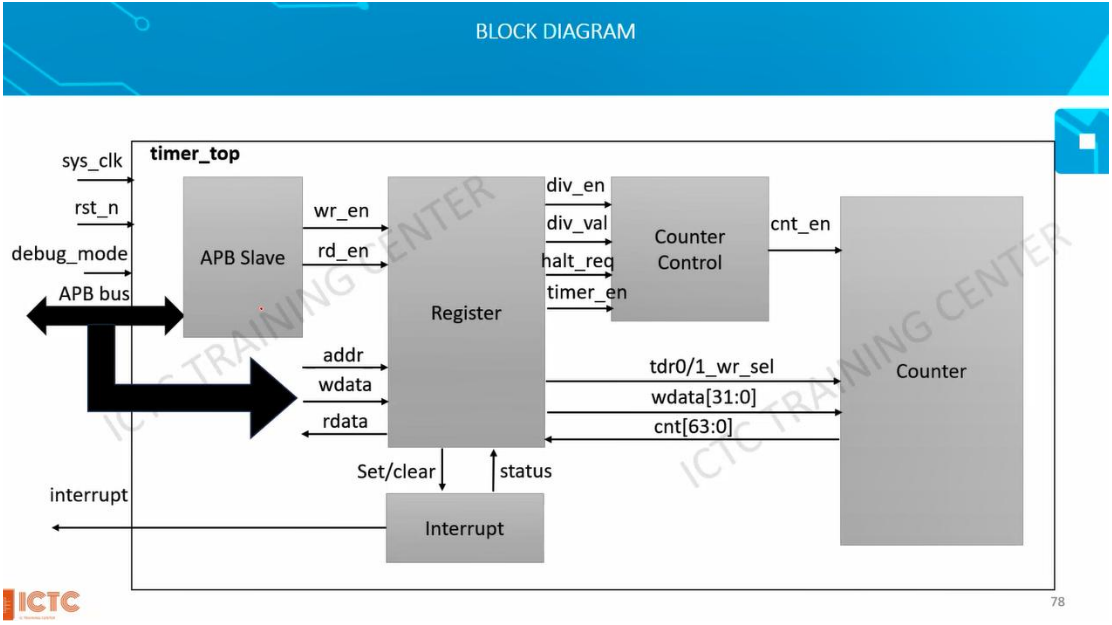
DESIGN NOTE

4. interrupt.
- interrupt pending (trigger condition): khi cnt = cmp
- interrupt status set: khi interrupt pending
- interrupt status clear: khi write 1 vào int_st
- interrupt output set: khi interrupt pending & interrupt is enable (int_en = 1)
- interrupt output clear: khi write 1 vào int_st hoặc disable interrupt

5. addr bit-wdith
- không phải lấy 32 bit address mà chỉ lấy số bit đủ để chứa address space thôi (cách tính như ss1)

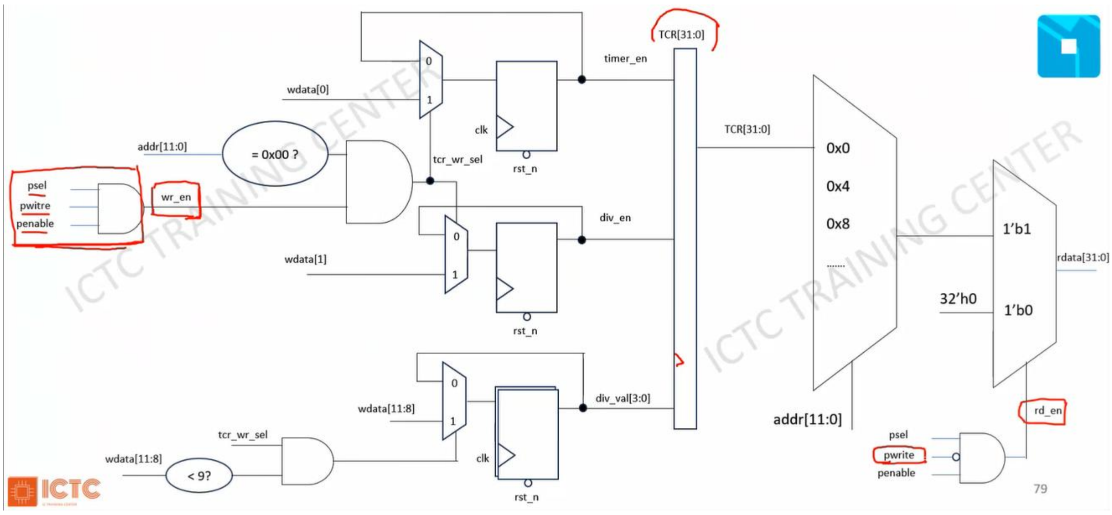
Block diagram

TIMER_TOP

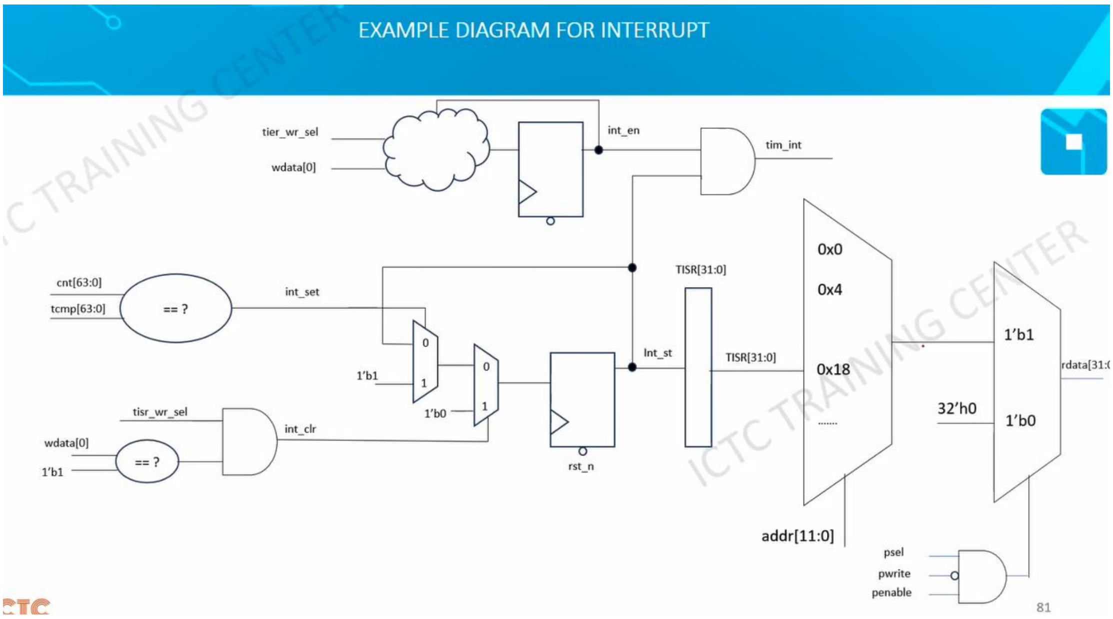


BLOCK DIAGRAM

TCR

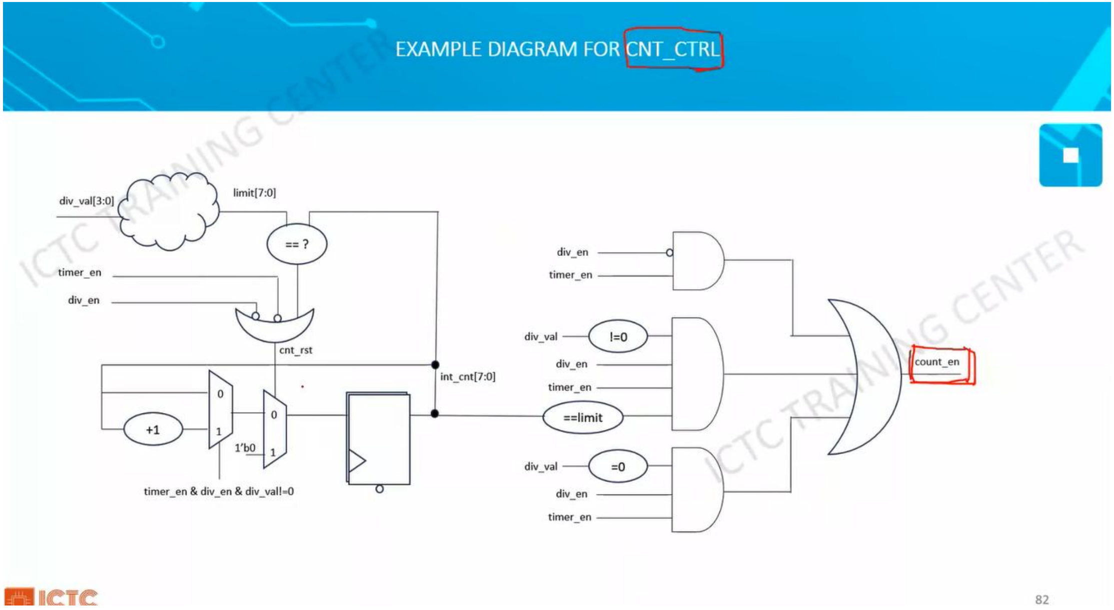


INTERUPT



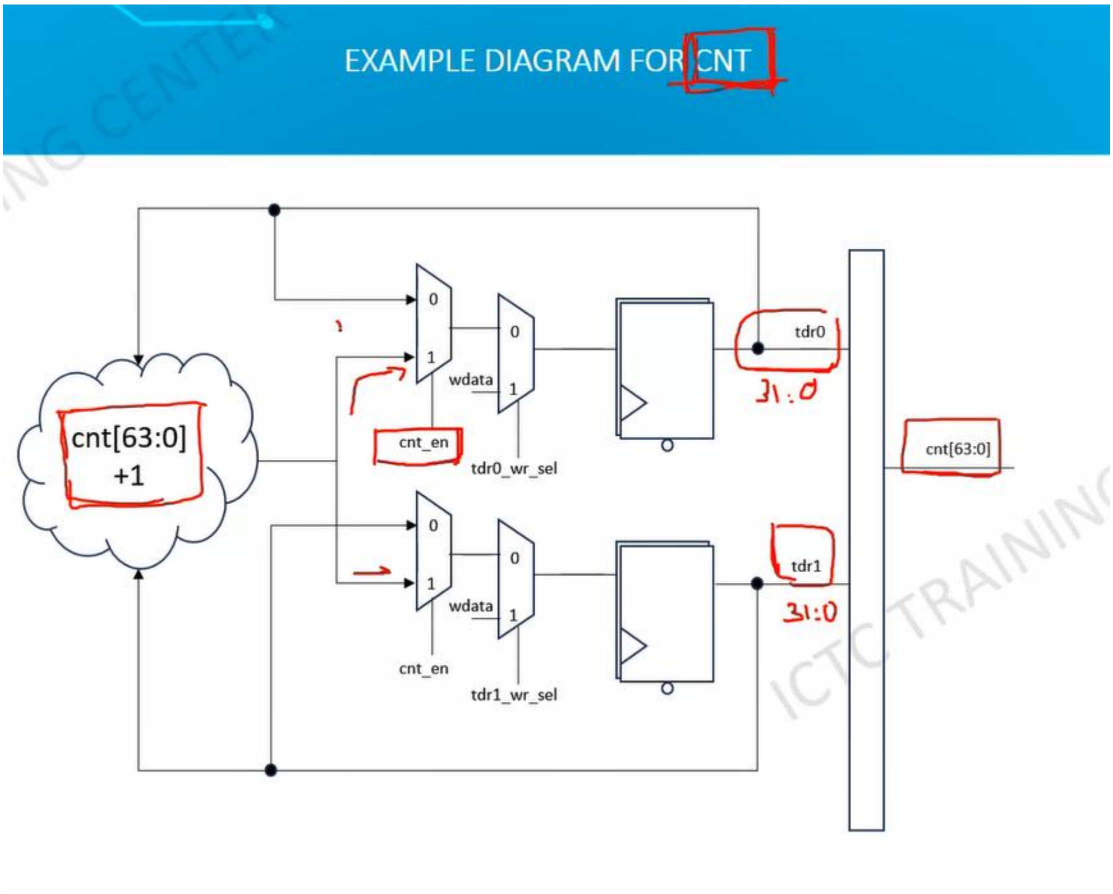
EXAMPLE DIAGRAM FOR INTERRUPT

COUNT_CONTROL



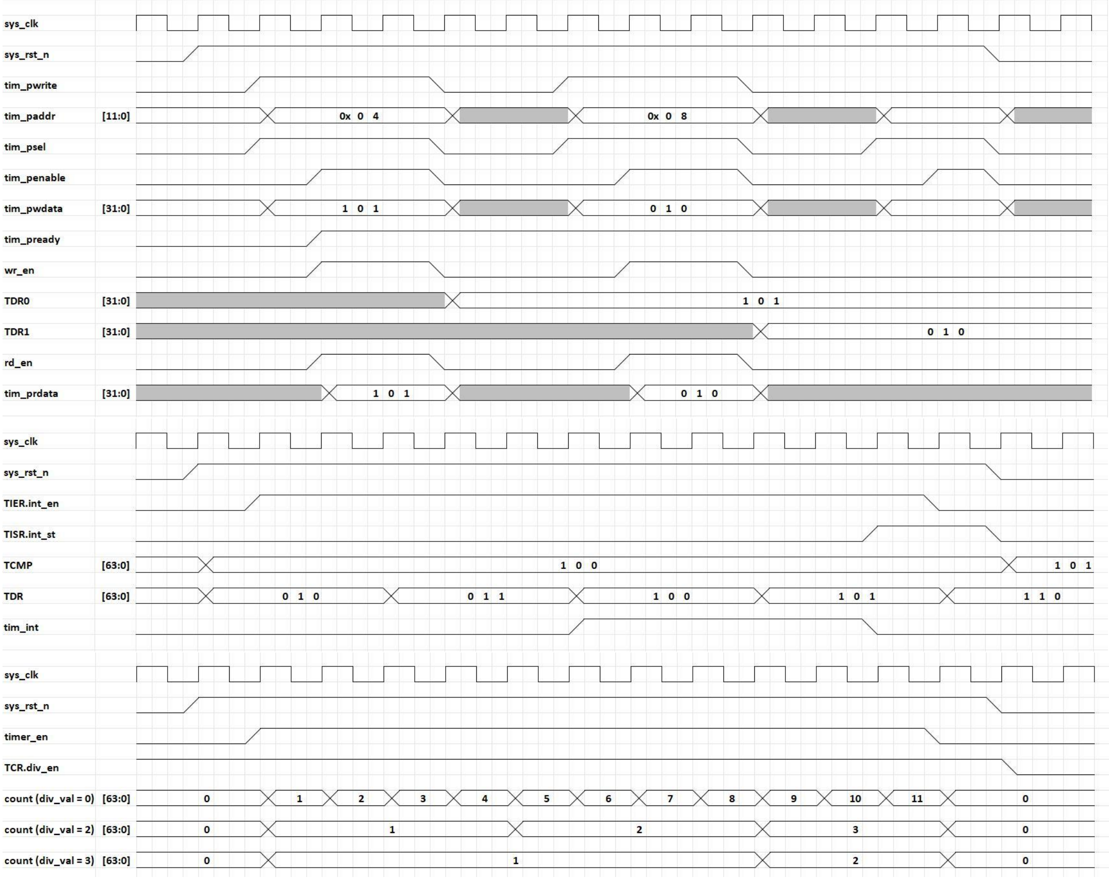
EXAMPLE DIAGRAM FOR CNT_CTRL

CNT



EXAMPLE DIAGRAM FOR CNT

WAVEFORM diagrams


sys_clk
sys_rst_n
tim_pwrite
tim_paddr [11:0]
0x 0 4
0x 0 8
tim_psel
tim_penable
tim_pwdata [31:0]
1 0 1
0 1 0
tim_pready
wr_en
TDR0 [31:0]
1 0 1
TDR1 [31:0]
0 1 0
rd_en
tim_prdata [31:0]
1 0 1
0 1 0
sys_clk
sys_rst_n
TIER.int_en
TISR.int_st
TCMP [63:0]
1 0 0
1 0 1
TDR [63:0]
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
tim_int
sys_clk
sys_rst_n
timer_en
TCR.div_en
count (div_val = 0) [63:0]
0
1
2
3
4
5
6
7
8
9
10
11
0
count (div_val = 2) [63:0]
0
1
2
3
0
count (div_val = 3) [63:0]
0
1
2
0